JOINT REPRESENTATION

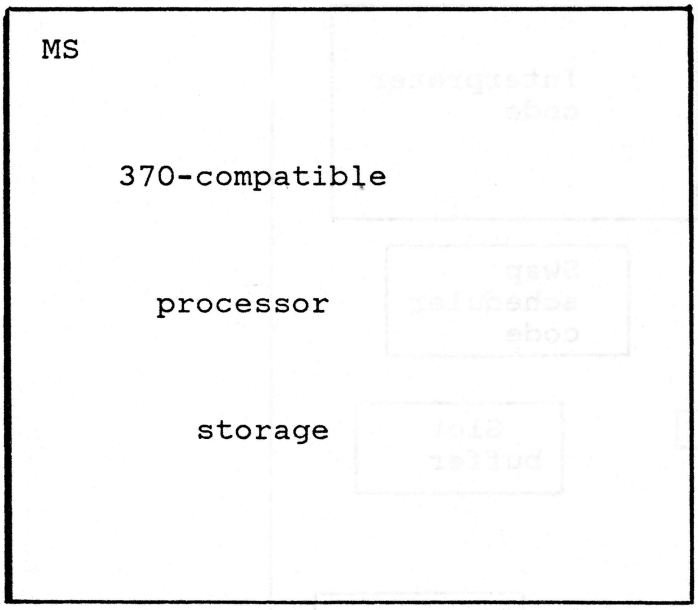
IBM 370-compatible computers represent main storage as a vector of characters (called bytes). Main Storage (MS) is usually accessed by scalar indices in range ⍳⍴MS.

Primitive machine operations are usually scalar in nature, via scalar indexed reference or assign.

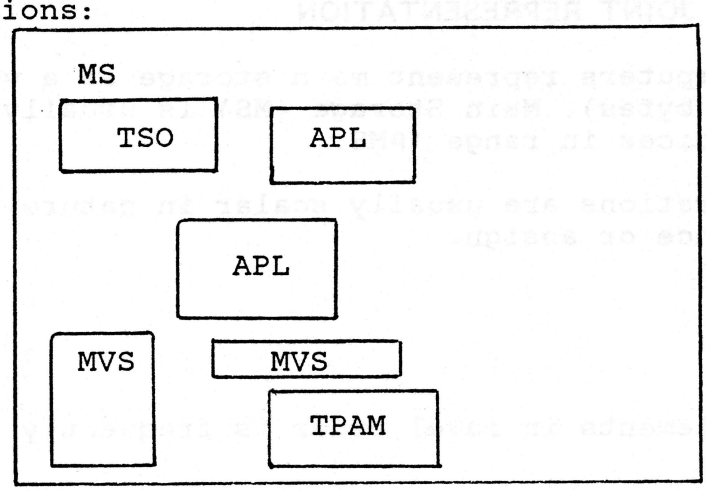
```
T←MS[⍵]

MS[⍵]←T
```

Iterative access to elements in ravel order is frequently faster than ordinal order.

If an entity in this implementation is stored in contiguous main store locations, it will be represented as a box drawn with solid lines:

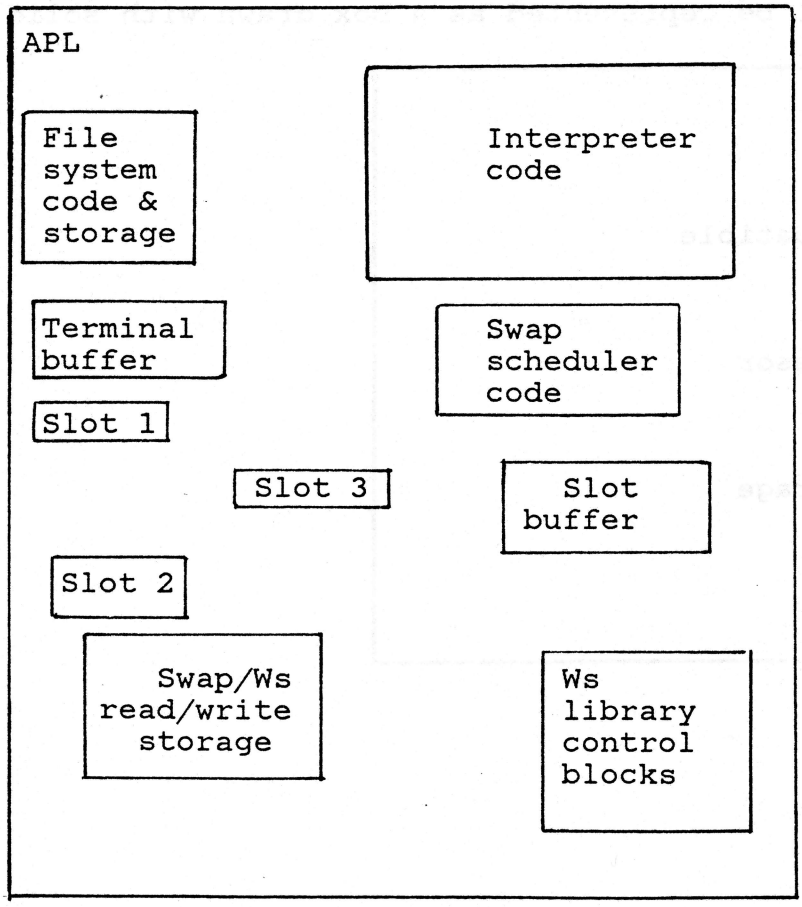


Usually storage on a machine is divided among several applications:



Note that an application may have several non-contiguous blocks of assigned storage.

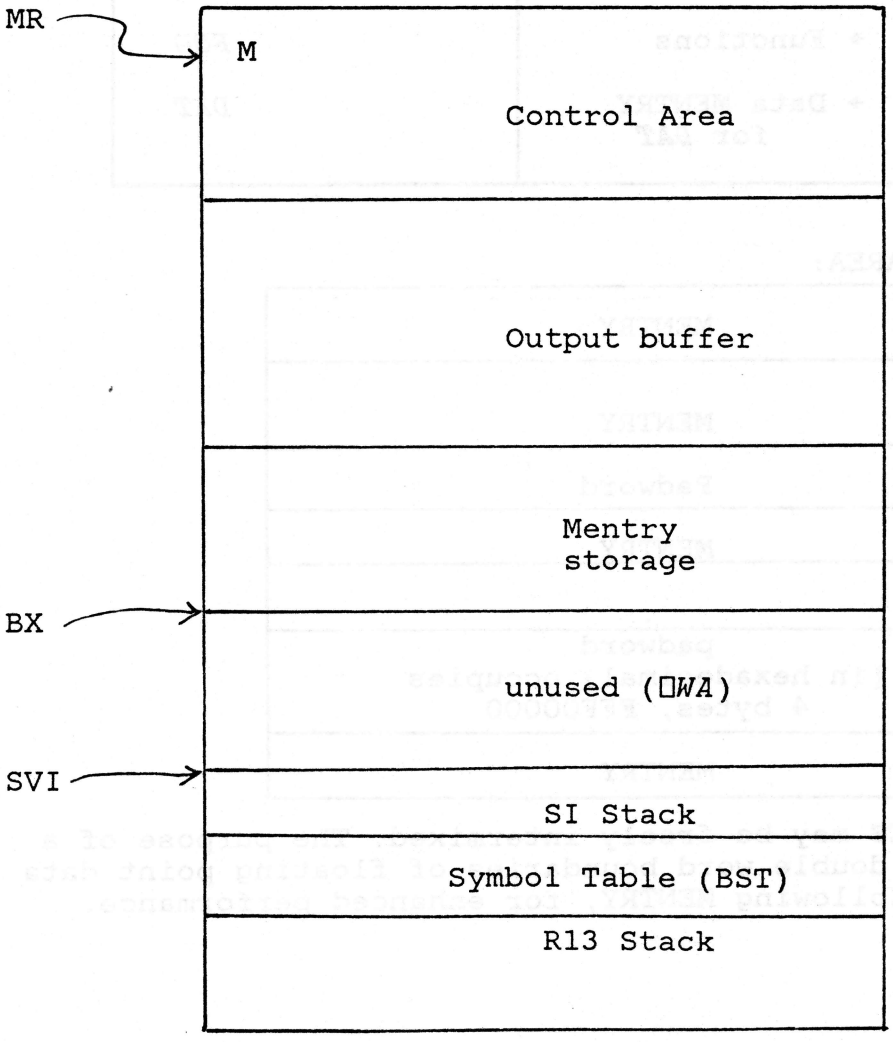
Storage allotted to APL is divided into several large components:



A slot is used to store an active workspace while processing occurs on it. A system typically has a number of slots much smaller than ⎕*UL*:
3 - 16 slots are currently allowed.

Since all active wss can't be in their own slots, they are moved to and from a mass storage known as a "swap device" on a demand basis, controlled by the "swap scheduler".

A Workspace Slot looks like:



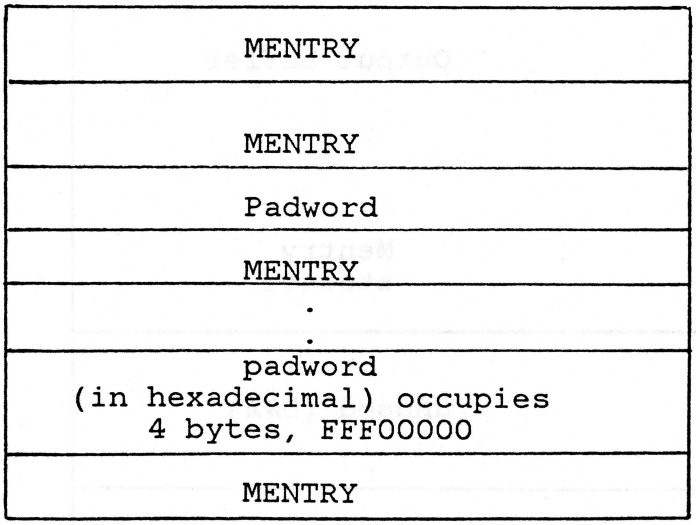
BASIC SYMBOL TABLE

A 3-column matrix associating a value, symbol class, and a symbol (identifier).

Typical Symbol Table entries:

Class	Value	Name
Variable	→ Data MENTRY for *Z*	*Z*
Function	→ Functions	*FOO*
Shared	→ Data MENTRY for *DAT*	*DAT*

THE MENTRY STORAGE AREA:

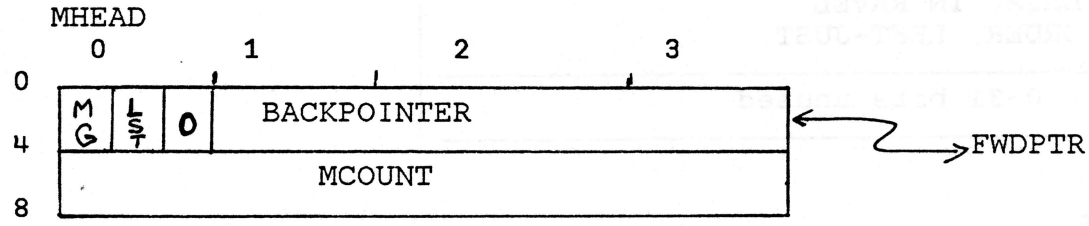


Padwords and MENTRIES may be freely intermixed. The purpose of a padword is to force double word boundaries of floating point data in the immediately following MENTRY, for enhanced performance.

THERE ARE SEVERAL TYPES OF MENTRIES:

- DATA (Boolean, integer, floating point, character, package, enclosed, complex)
- CODESTRING (Function line, IMMEX line, Execute line)
- LIST (Function directory, group, □*FMT* or index list, operator frame)
- LONG PRINTNAME
- GARBAGE MENTRY

COMMON MENTRY HEADER:



The first 8 bytes of an MENTRY have a common format:

- The first bit (MGBIT) is *X'80'* if the MENTRY may be discarded.
- MLSTBIT is *X'40'* if the MENTRY is a list MENTRY.
- The BACKPOINTER is a byte address (called an M-relative address). The word addressed by the BACKPTR is called the FORWARD POINTER and must contain the M-relative address of the first byte of the MENTRY.

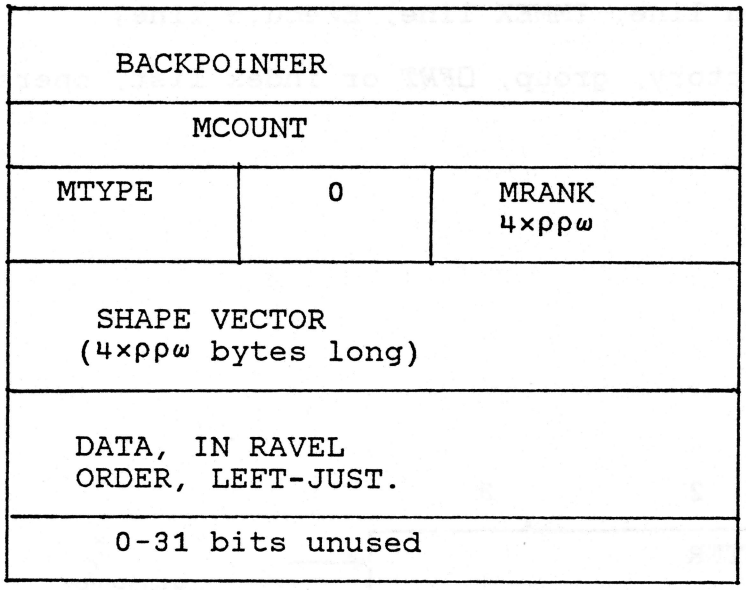
This linkage allows arbitrary movement of the MENTRY or its FWDPTR in the WS, thereby facilitating WS storage management.

- MCOUNT is a Fullword (4 bytes) containing the number of bytes in the MENTRY.

DATA MENTRIES

Data MENTRIES are used to store arrays of APL (& packages). They have a common format:

MENTRY ω



MTYPE IS

BOOLEAN	TYPB 1
INTEGER	TYPI 2
FLOATING	TYPF 3
CHARACTER	TYPC 4
PACKAGE	TYPP 5
ENCLOSED ARRAY	TYPA 6
COMPLEX	TYPZ 7

MRANK is four times the array rank, so that MRANK is the number of bytes occupied by the shape vector.

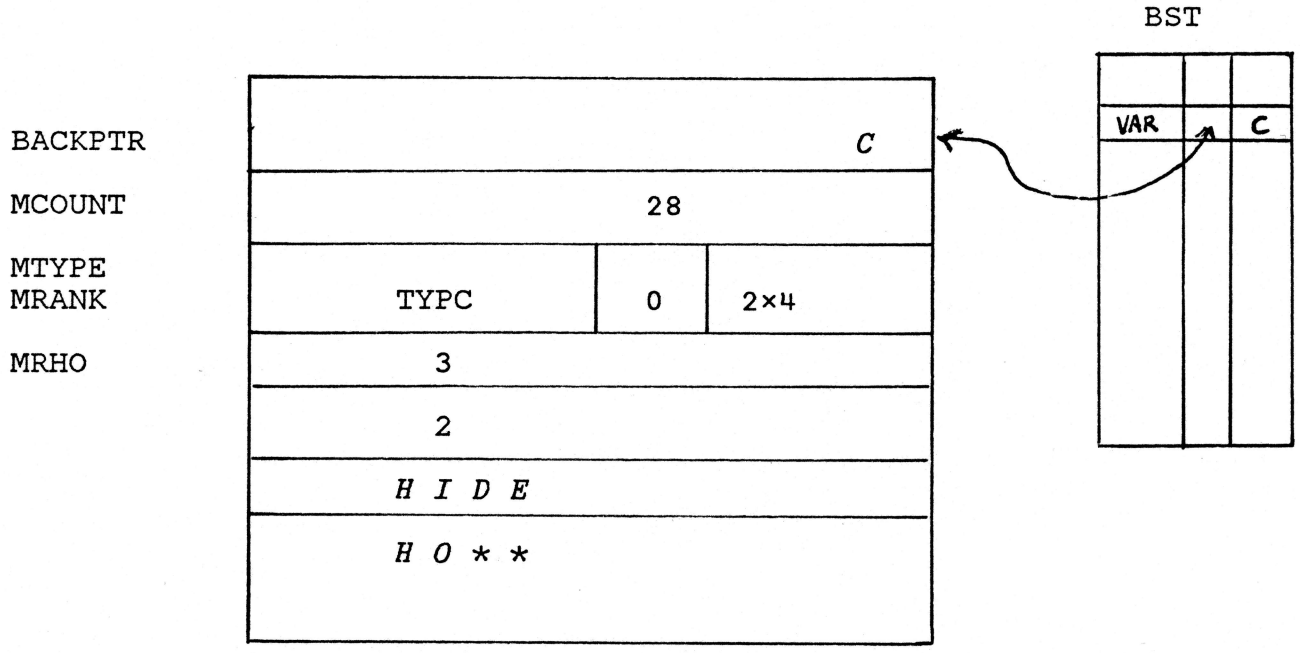
The shape vector is a vector of 0 or more fullword integers representing the shape vector of the array.

DATA ELEMENTS ARE REPRESENTED AS FOLLOWS:

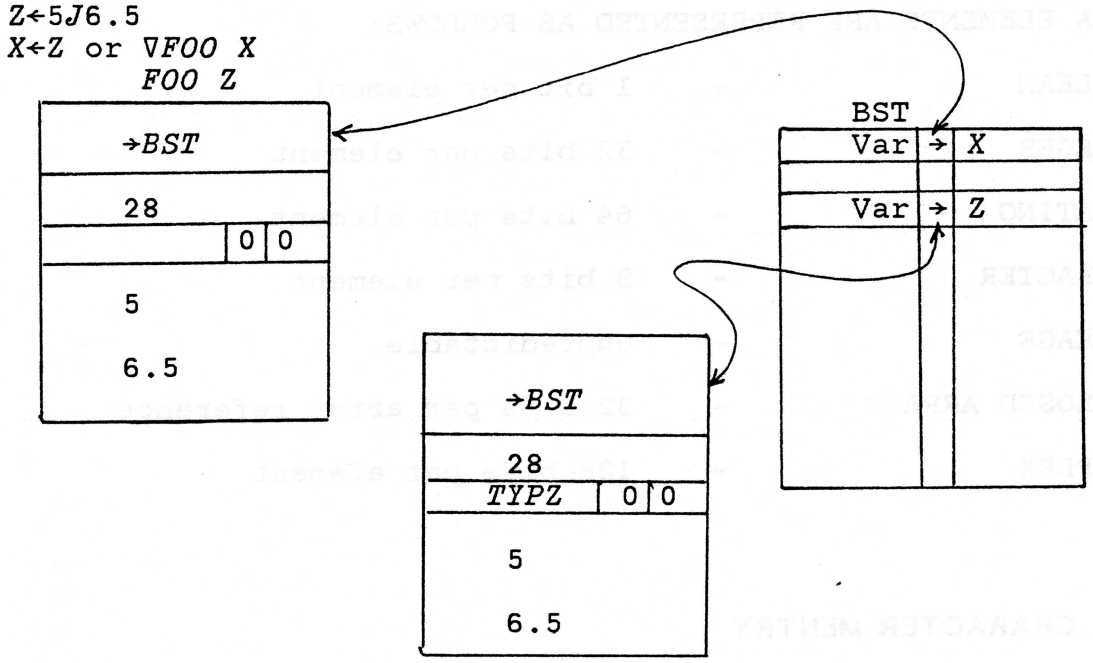
BOOLEAN	-	1 bit per element
INTEGER	-	32 bits per element
FLOATING	-	64 bits per element
CHARACTER	-	8 bits per element
PACKAGE	-	Unpredictable
ENCLOSED ARRAYS	-	32 bits per array reference
COMPLEX	-	128 bits per element

CHARACTER MENTRY

```
 □← C← 3 2⍴'HIDEHO'
HI
DE
HO
```



CONVENTIONAL ASSIGNMENT

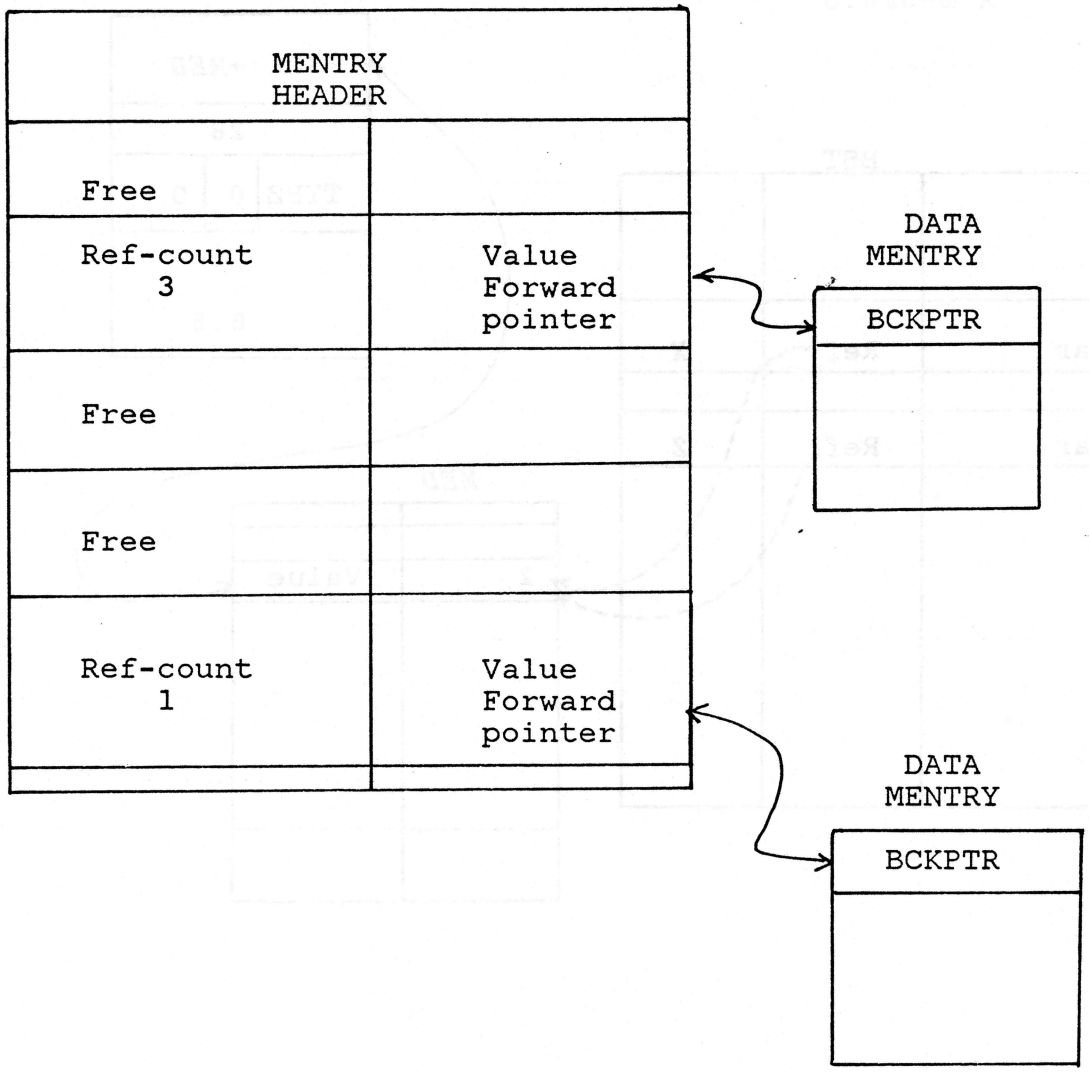


CPU time required to physically copy MENTRY.

SPACE required to physically copy the MENTRY.

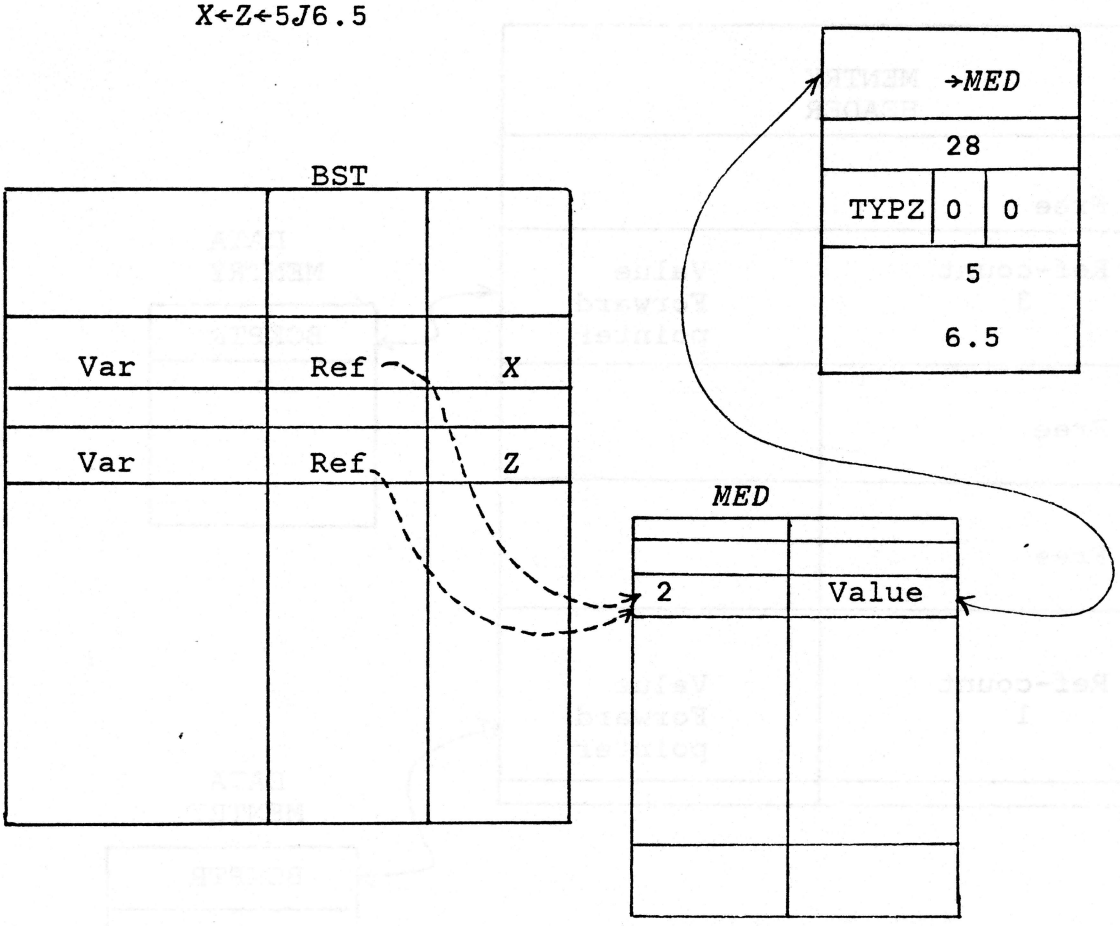
MENTRY DIRECTORY

2-column matrix of reference counts and MENTRY addresses.

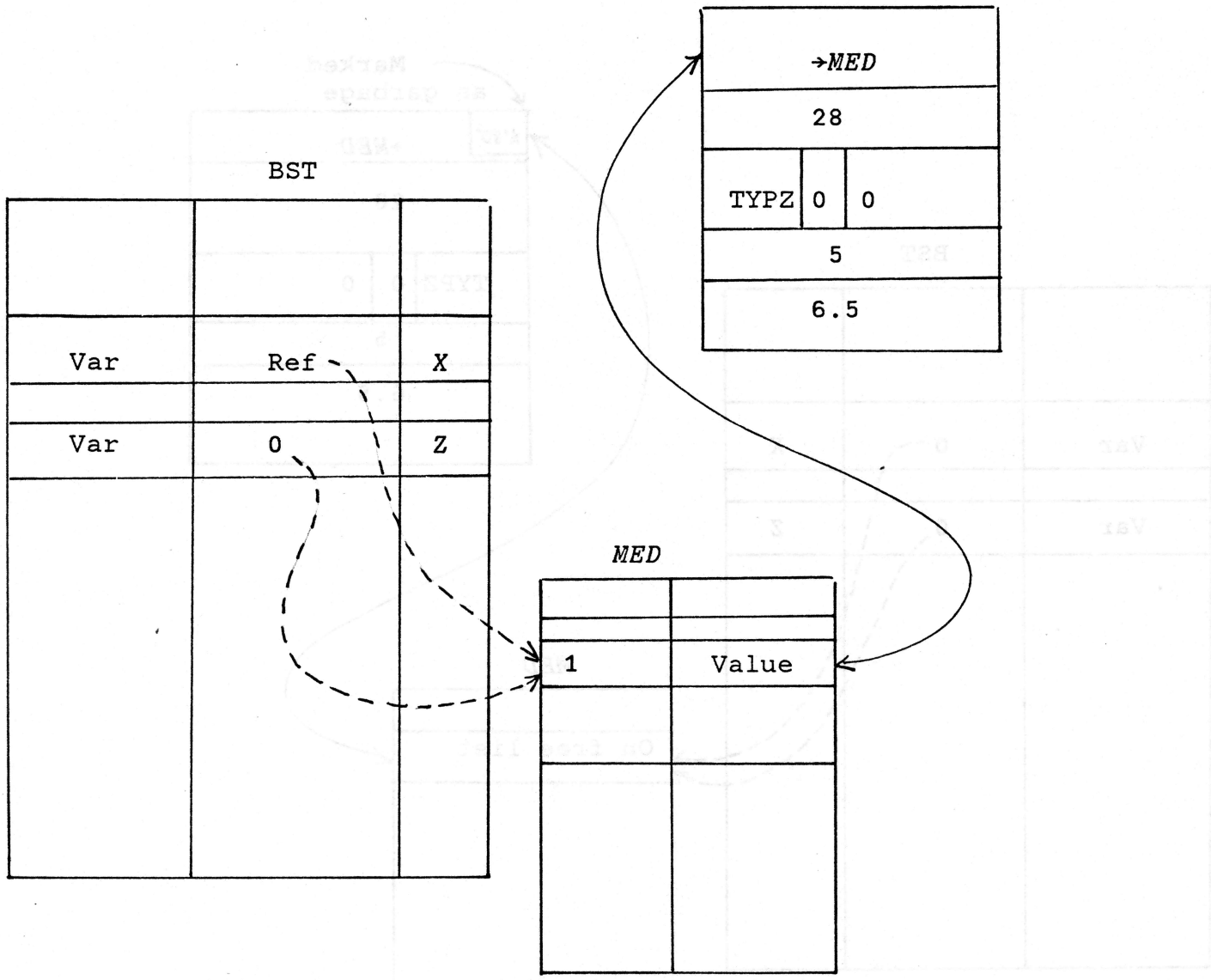


JOINT REPRESENTATION OF ASSIGNMENT

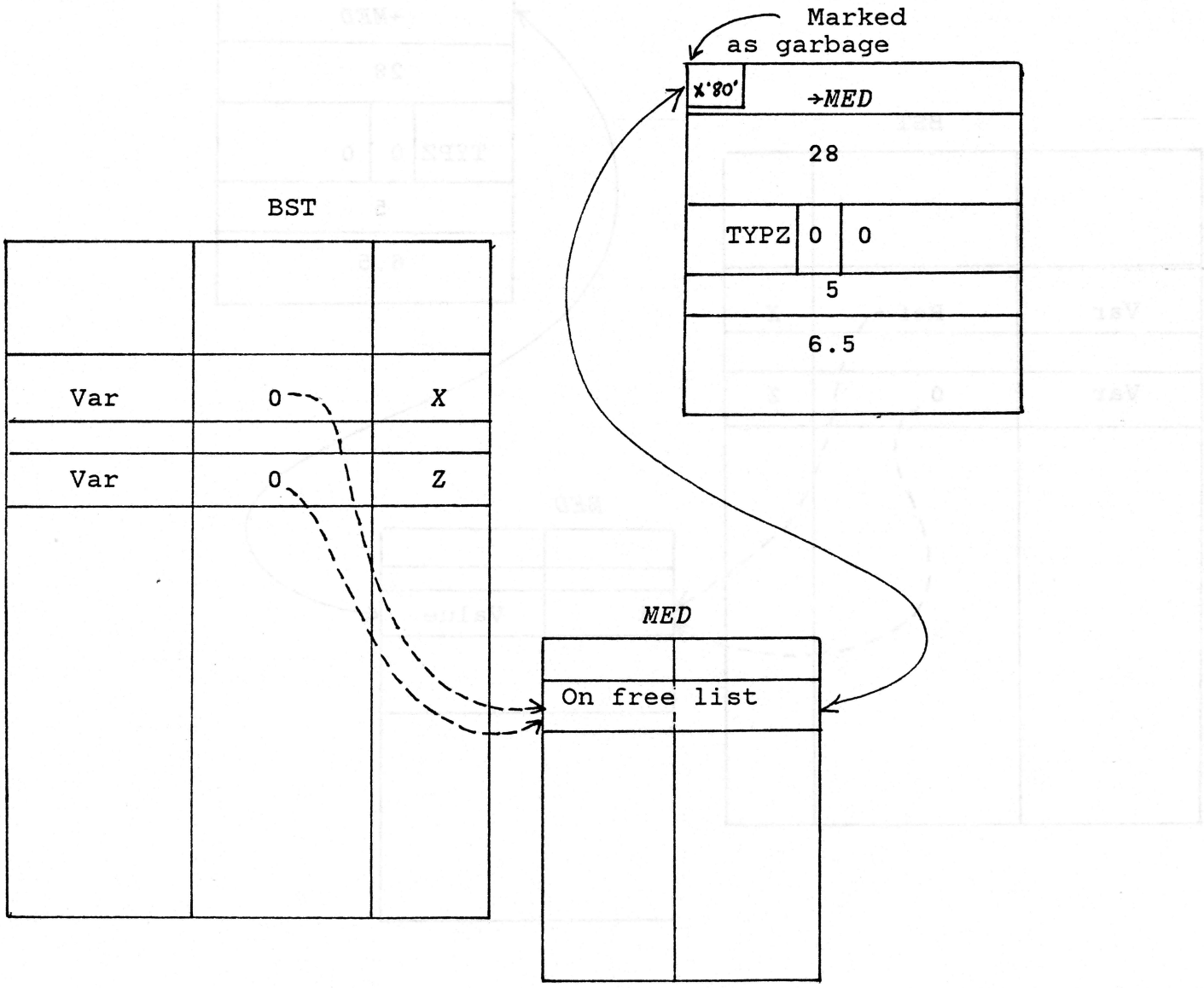
X←Z←5J6.5



```
⎕EX 'Z'
```



```
⎕EX 2 1⍴'XZ'
```

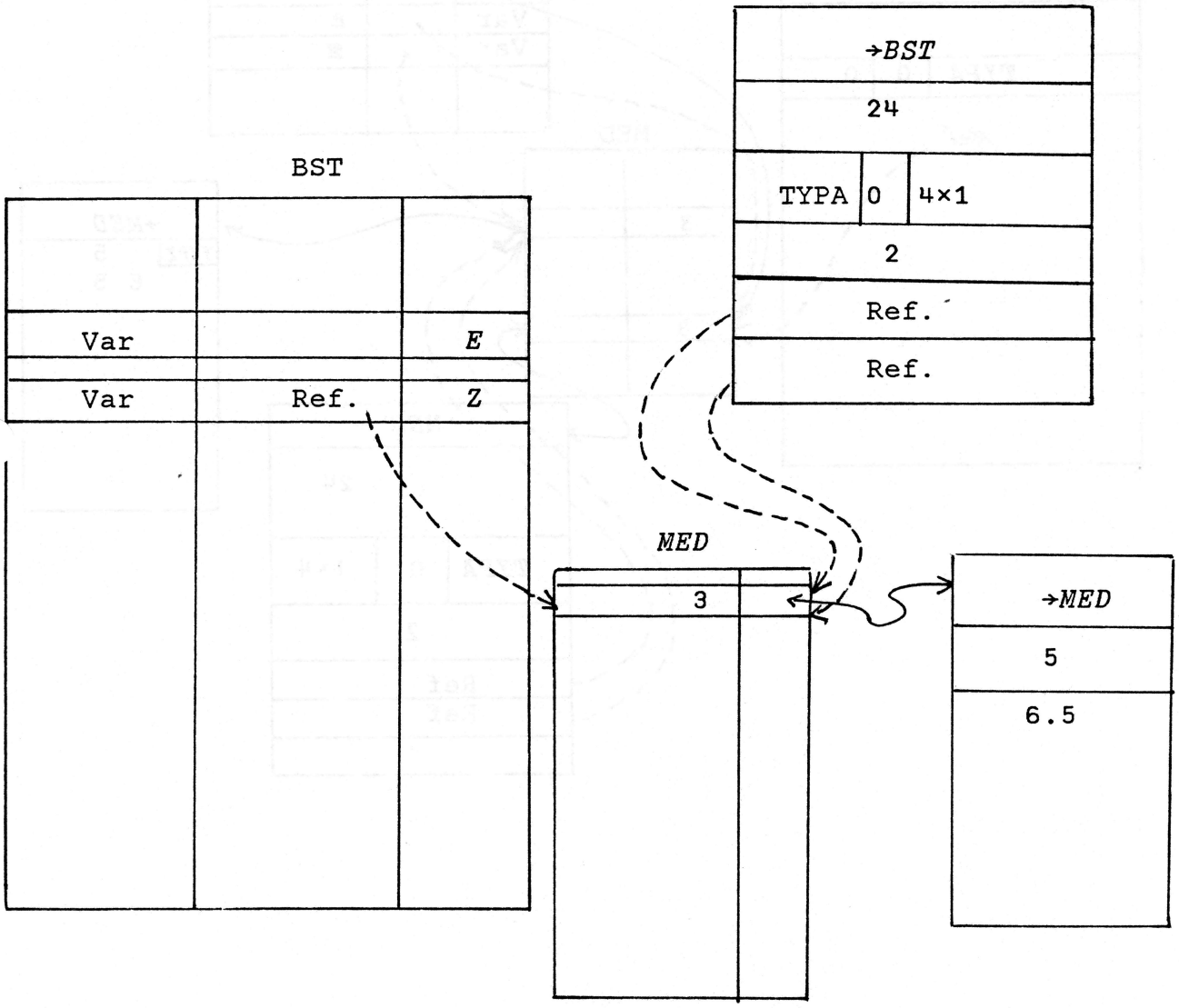

Marked
as garbage
X'80'
→MED
28
TYPZ
0
0
5
6.5
BST
Var
0
X
Var
0
Z
MED
On free list

REFERENCE

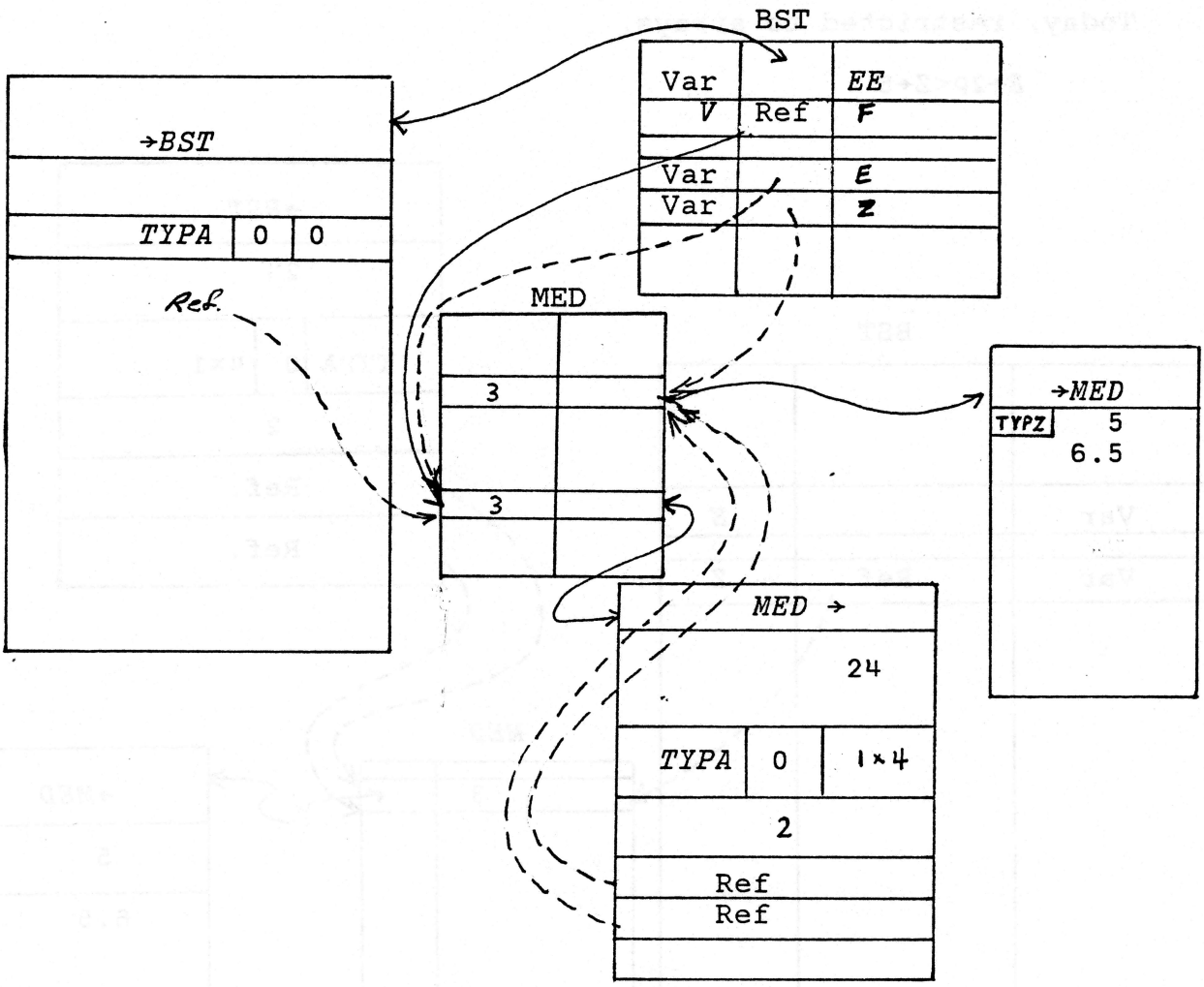
A fullword token representing an APL object.

Today, restricted to arrays.

E←2ρ<*Z*←5



EE←<E
F←E


BST
Var
EE
V
Ref
F
Var
E
Var
Z
→BST
TYPA
0
0
Ref
MED
3
3
→MED
TYPZ
5
6.5
MED →
24
TYPA
0
1×4
2
Ref
Ref

EVALUATION OF VARIABLES

Previously, NAME was stacked, so that

```
(A←2)+A←4
```

gave the wrong result.

```
A←4
(A←2)+A
(2)+A
2+2
```

Now the reference to the value is stacked: a value is detached from a name at time of encounter by synyax analyzer.

```
(A←2)+A←4
(A←2)+4
(2)+4
2+4
```

An argument or a result of a function may be a reference. This allows some identities to run fast and requires no space for a result MENTRY:

```
,VECTOR
(ρω)ρω
(ρω)↑ω
((ρρω)ρ0)↓ω
1/ω  (ω not scalar)
>SIMPLE
>SCALAR
⍕CHARACTER
...
...
```

FUTURE

Joint representation extended to functions and operators.

This will allow:

1. Function to expunge or redefine itself (recursion will invoke new definition).

2. Allow (⎕EX'FN')FN ⍵

3. SI DAMAGE will not occur. SI WARNING if function on stack is not the same as its ⎕CR.